UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/666,818	09/17/2003	Theron Tock	11389-035	8933

20583	7590	03/14/2006

JONES DAY
222 EAST 41ST ST
NEW YORK, NY 10017

EXAMINER
WON, MICHAEL YOUNG

ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 03/14/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	Applicant(s)
Office Action Summary	10/666,818	TOCK, THERON
	Examiner	Art Unit	
	Michael Y. Won	2155	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>23 December 2005</u>.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-60</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-11,13-41 and 43-60</u> is/are rejected.

7)☒ Claim(s) <u>12 and 42</u> is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____ .

DETAILED ACTION

1.   This action is in response to the amendment filed December 23, 2005.

2.   Claims 1 and 31 have been amended.

3.   Claims 1-60 have been examined and are pending with this action.

Claim Rejections - 35 USC § 112

4.   Claims 1 and 31 previously rejected under 35 U.S.C. 112 second paragraph has

been withdrawn responsive to the amendment.

Response to Arguments

5.   Applicant's arguments with respect to claim1 and 31 have been considered but

are moot in view of the new ground(s) of rejection.

A. The applicant(s) argued that the reference U.S. Pat. No. 6,006,269 ("Phaal")

does not explicitly teach or suggest that the scheduler is located at the client side.

Based on the new rejection as set forth below, this argument is moot.

B. The applicant(s) assert that Phaal does not explicitly teach or suggest "the client-side code registering a request entry with a client-side scheduler while the network connection is unavailable".

The registration of a request is implicit since Phaal teaches of that system is session-based. Furthermore, the combinational teachings of U.S. Pat. No. 6,934,735 ("Emens"), teachings the client-side scheduler, and Phaal, clearly teach this limitation.

C. The applicant(s) assert that Phaal does not explicitly teach or suggest "storing the request entry until a network connection is available for use by the client-side code".

In response, see the newly cited reference location(s) provided in the rejection below, which better teaches the limitation.

D. The applicant(s) assert that Phaal does not explicitly teach or suggest "the client-side scheduler notifying the client-side code that the network connection is available".

In response, see the newly cited reference location(s) provided in the rejection below, which better teaches the limitation.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6.   Claims 1-3, 20, 22-25, 28-33, 50, 52-55, and 58-60 are rejected under 35 U.S.C.

103(a) as being unpatentable over Phaal (US 6,006,269 A) in view of Emens et al. (US

6,934,735 B1)

INDEPENDENT:

As per **claim 1**, Phaal teaches a method for sending a request from a client

computer to a server computer (see col.1, lines 12-18), wherein the client computer is

connected to the server computer via a network (see col.5, lines 17-20), where the

server computer runs server-side code associated with a client/server application,

wherein the client computer runs client-side code associated with the client/server

application (see col.4, lines 38-42; col.5, lines 47-49; and col.14, lines 55-60), the

method comprising;

the client-side code registering a request entry while the network connection is

unavailable (implicit: see col.2, lines 35-42: "deferred clients... session-based"; col.4,

lines 46-53; and col.4, line 58-col.5, line 6), wherein the request entry represents the

request to be sent from the client computer to the server computer (see col.4, lines 46-

48: "normally admits messages to a server");

storing the request entry until a network connection is available for use by the

client-side code (see col.7, lines 12-32; col.12, lines 46-51 and col.15, lines 56-63);

notifying the client-side code that the network connection is available (see col.4,

lines 53-57 and col.7, lines 18-22);

the client-side code using the network connection to send the request to the

server computer (see col.5, lines 17-20).

Although Phaal teaches of a scheduler, Phaal does not explicitly teach that the

scheduler is client side (see col.3, lines 17-18: "client resident software for use in

accessing a host" and col.12, lines 34-37 & 42-51).

Emens teaches of a client-side scheduler (see Fig.3, #324 and col.4, lines 9-16 &

36-39).

It would have been obvious to a person of ordinary skill in the art at the time the

invention was made to employ the teaching of Emens within the system of Phaal by

implementing a client-side scheduler within the method for sending a request from a

client computer to a server computer because such implementation would allow for no

connection to be made to the server initially which further reduces overloading and

bottlenecks and because Phaal suggests that "Other mechanisms for assisting

admissions control on the client side may be used" (see col.14, lines 60-62).

As per *claim 31*, Phaal teaches a system for enabling a client/server application,

the system comprising:

a server computer (see col.1, line 12), wherein the server computer runs

server-side code associated with the application (see col.4, lines 38-42; col.5 lines 47-

49; and col.14, lines 55-60);

a client computer (see col.1, line 14) connected to the server computer via a

network, wherein the client computer runs client-side code associated with the

application (see col.4, lines 38-42; col.5, lines 47-49; and col.14, lines 55-60);

wherein the client-side code is operable to interface a scheduler in order to

manage requests to the server computer (see col.6, lines 22-49);

wherein the client-side code is executable to register a request entry with the

scheduler while the network connection is unavailable (implicit: see col.2, lines 35-42:

"deferred clients... session-based"; col.4, lines 46-53; and col.4, line 58-col.5, line 6),

wherein the request entry represents a request that the client-side code needs to send

to the server computer (see col.4, lines 46-57);

wherein the scheduler is executable to store the request entry until a network

connection is available for use by the client-side code (see col.7, lines 12-32; col.12,

lines 46-51 and col.15, lines 56-63);

wherein the scheduler is executable to notify the client-side code that the network

connection is available (see col.4, lines 53-57 and col.7, lines 18-22);

wherein the client-side code is executable to use the network connection to send

the request to the server computer (see col.5, lines 17-20).

Although Phaal teaches of a scheduler, Phaal does not explicitly teach that the scheduler is client side (see col.3, lines 17-18: "client resident software for use in accessing a host" and col.12, lines 34-37 & 42-51).

Emens teaches of a client-side scheduler (see Fig.3, #324 and col.4, lines 9-16 & 36-39).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teaching of Emens within the system of Phaal by implementing a client-side scheduler within the system for enabling a client/server application because such implementation would allow for no connection to be made to the server initially which further reduces overloading and bottlenecks and because Phaal suggests that "Other mechanisms for assisting admissions control on the client side may be used" (see col.14, lines 60-62).

DEPENDENT:

As per *claims 2 and 32*, Phaal further teaches wherein said client-side code registering a request entry with the scheduler comprises the client-side code informing the scheduler of a callback routine to associate with the request entry; wherein said scheduler storing the request entry comprises the scheduler storing information identifying the callback routine; wherein said scheduler notifying the client-side code that the network connection is available comprises the scheduler invoking the callback routine (see col.14, lines 55-60).

As per *claims 3 and 33*, Phaal further teaches wherein said client-side code registering a request entry with the scheduler comprises the client-side code storing request context information for the request entry (see col.12, lines 46-51 and col.15, lines 56-63); wherein said client-side code using the network connection to send the request to the server computer comprises the client-side code retrieving the request context information for the request entry and using the request context information to formulate the appropriate request to send to the server computer (see col.11, lines 21-30 and col.12, lines 23-26).

As per *claims 20 and 50*, Phaal further teaches wherein the scheduler is an independent software component (see col.4, lines 38-42 and col.6, lines 48-49).

As per *claims 22, 23, 52, and 53*, Phaal further teaches wherein the scheduler is not an independent software component, but rather is tightly integrated with the client-side code and the scheduler executes on the client computer (see col.4, lines 36-38 and col.11, line 18).

As per *claims 24 and 54*, Phaal further teaches wherein the scheduler executes on a computer connected to the client computer via a local area network (LAN) (see col.4, lines 36-42 and col.7, lines 36-41).

As per *claims 25 and 55*, Phaal further teaches wherein the client/server application is an Internet application, wherein the network is the Internet (see col.1, lines 14-15 & 20-23, and col.7, lines 33-41).

As per *claims 28 and 58*, Phaal further teaches wherein the request sent by the

client-side code references a resource accessible over the Internet via a uniform

resource locator (URL) (see col.7, lines 8-11 & 15-17; and col.9, line 66-col.10 lines 2).

As per *claims 29, 30, 59, and 60*, Phaal further teaches wherein the client-side

code interfaces with the Web browser code base in order to send the request over the

Internet; and wherein the client-side code uses the Web browser control as an

embedded component (see col.13, lines 15-20).

7.   Claims 4-8, 10, 18, 34-38, 40, and 48 are rejected under 35 U.S.C. 103(a) as

being unpatentable over Phaal (US 6,006,269 A) and Emens et al. (US 6,934,735 B1),

and further in view of Griffin et al. (US 4,949,251 A).

As per *claims 4 and 34*, Phaal and Emens teaches all the limitations except

wherein said client-side code storing request context information for the request entry

comprises the client-side code storing the request context information in a database.

Griffin teaches of a client storing a request in a database (see Fig.1, #19 and col.6, lines

33-50).

It would have been obvious to a person of ordinary skill in the art at the time the

invention was made to employ the teachings of Griffin within the system of Phaal and

Emens by implementing a client-side code storing request context information in a

database within the client-server offline system because Griffin teaches that such an

implementation enhances the availability of the system by enabling other processors to continue processing when one or more processors fail (see Griffin: col.6, lines 51-59).

As per *claims 5 and 35*, Phaal further teaches wherein said client-side code registering a request entry with the scheduler comprises the client-side code informing the scheduler of a database key identifier for the request context information stored in the database; and wherein said scheduler storing, notifying client-side code, and retrieving the request context information comprises the scheduler storing, passing, and using the database key identifier (see col.2, line 64-col.3 line 3; col.10, lines 3-21; and col.12, lines 32-34).

As per *claims 6 and 36*, Phaal and Emens teaches all the limitations except wherein the request that the client-side code needs to send to the server computer requires exactly-once semantics; wherein said client-side code registering a request entry with the scheduler comprises the client-side code informing the scheduler that the request requires exactly once semantics; wherein said scheduler storing the request entry comprises the scheduler storing information specifying that the request requires exactly-once semantics; wherein, in the case of an error during said client-side code using the network connection to send the request to the server computer, the method includes means for providing error recovery for the request, wherein said means for providing error recovery ensure that the server computer does not perform the request more than once.

Griffin teaches wherein the request that the client-side code needs to send to the server computer requires exactly-once semantics; wherein said client-side code

registering a request entry with the scheduler comprises the client-side code informing

the scheduler that the request requires exactly once semantics; wherein said scheduler

storing the request entry comprises the scheduler storing information specifying that the

request requires exactly-once semantics (see title); wherein, in the case of an error

during said client-side code using the network connection to send the request to the

server computer, the method includes means for providing error recovery for the

request (see col.3, lines 39-42), wherein said means for providing error recovery ensure

that the server computer does not perform the request more than once (see col.4, lines

31-33 & 54-56).

It would have been obvious to a person of ordinary skill in the art at the time the

invention was made to employ the teachings of Griffin within the system of Phaal and

Emens by implementing exactly-once semantics and error recovery within the client-

server offline system because this enables consistent and current data integrity in the

event of a failure within any node of the system.

As per *claims 7 and 37*, Phaal and Emens teaches all the limitations except

wherein said error during the client-side code using the network connection to send the

request to the server computer comprises the network connection being terminated

before the response to the request is received from the server computer; wherein said

means for providing error recovery for the request comprise: the client-side code

informing the scheduler that the request failed; the scheduler marking the request entry

corresponding to the request with a status indicating that the request needs error

recovery, in response to being informed by the client-side code that the request failed; a

network connection being re-established; the scheduler using the network connection to

query the server computer in order to determine whether the request was successfully

completed; the scheduler informing the client-side code that the request was

successfully completed if the scheduler determines from the query to the server

computer that the request was successfully completed; the scheduler marking the

request entry corresponding to the request with a status indicating that the request

should be re-sent if the scheduler determines from the query to the server computer that

the request was not successfully completed.

Griffin teaches wherein said error comprises the network connection being

terminated before the response to the request is received (see col.11, lines 58-60);

wherein said means for providing error recovery for the request comprise: informing that

the request failed (see col.7, lines 7-9); a network connection being re-established;

using the network connection to query, in order to determine whether the request was

successfully completed; informing that the request was successfully completed if the

request was successfully completed (see col.9, lines 49-52); marking the request entry

corresponding to the request with a status indicating that the request should be re-sent

upon determining from the query that the request was not successfully completed (see

col.9, lines 52-57).

It would have been obvious to a person of ordinary skill in the art at the time the

invention was made to employ the teachings of Griffin within the system of Phaal and

Emens by implementing an error recovery within the client-server offline system

because this enables consistent and current data integrity in the event of a failure within

any node of the system. Also in order to maintain data consistency, such error recovery

system can be implemented between all interconnected nodes. Therefore whatever

devices resides at each node, an error recovery system can be implemented between

said nodes.

As per *claims 8 and 38*, Phaal and Emens teaches all the limitations except

wherein the request sent by the client-side code to the server computer includes a job

ID for the request; wherein, in response to successfully performing the request, the

server computer is operable to log the job ID for the request; wherein said scheduler

querying the server computer in order to determine whether the request was

successfully completed comprises the scheduler querying the server computer in order

to determine whether the server computer logged the job ID for the request.

Griffin teaches wherein the request sent by the client-side code to the server

computer includes a job ID for the request; wherein, in response to successfully

performing the request, the server computer is operable to log the job ID for the request;

wherein said scheduler querying the server computer in order to determine whether the

request was successfully completed comprises the scheduler querying the server

computer in order to determine whether the server computer logged the job ID for the

request (see col.3, lines 42-51; col.4, lines 36-45; and col.9, lines 49-55).

It would have been obvious to a person of ordinary skill in the art at the time the

invention was made to employ the teachings of Griffin within the system of Phaal and

Emens by implementing a job ID for requests within the client-server offline system

because this would allow the system to keep track of what data request has been

completed and what data is most current in the event of a system failure to ensure

exactly-once semantics.

As per *claims 10 and 40*, Phaal and Emens teaches all the limitations, except

wherein the client-side cache is implemented using a database. Griffin teaches wherein

the client-side cache is implemented using a database (see Fig.1, #19 and col.6, lines

33-50).

It would have been obvious to a person of ordinary skill in the art at the time the

invention was made to employ the teachings of Griffin within the system of Phaal and

Emens by implementing a client-side cache implementing a database within the client-

server offline system because Griffin teaches that such an implementation enhances

the availability of the system by enabling other processors to continue processing when

one or more processors fail (see Griffin: col.6, lines 51-59).

As per *claims 18 and 48*, Phaal and Emens teaches all the limitations except

wherein the method comprises error recovery means in the event of abnormal

termination of the client-side code; wherein said error recovery means comprise

ensuring that upon restart of the client-side code, any requests requiring exactly-once

semantics that were in progress when the abnormal termination of the client-side code

occurred are not performed multiple times.

Griffin teaches wherein the method comprises error recovery means in the event

of abnormal termination of the client-side code; wherein said error recovery means

comprise ensuring that upon restart of the client-side code, any requests requiring

exactly-once semantics that were in progress when the abnormal termination of the

client-side code occurred are not performed multiple times (see col.4, lines 31-33 & 54-56).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Griffin within the system of Phaal and Emens by implementing exactly-once semantics and error recovery within the client-server offline system because this enables consistent and current data integrity in the event of a failure within any node of the system.

8.   Claims 9, 11, 13-17, 19, 21, 39, 41, 43-47, 49, and 51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Phaal (US 6,006,269 A) and Emens et al. (US 6,934,735 B1), and further in view of Braddy (US 6,141,759 A).

As per **claims 9 and 39**, Phaal and Emens teaches all the limitations except wherein the client-side code maintains a client-side cache; wherein, in response to determining a need for particular information, the client-side code is operable to check the client-side cache for the needed information; wherein, if the client-side cache comprises the needed information, the client-side code is operable to retrieve the needed information from the client-side cache; wherein, if the client-side cache does not comprise the needed information, the client-side code is operable to register a request entry with the scheduler, wherein the request entry represents a request to retrieve the needed information from the server computer.

Braddy teaches wherein the client-side code maintains a client-side cache; wherein, in response to determining a need for particular information, the client-side

code is operable to check the client-side cache for the needed information; wherein, if the client-side cache comprises the needed information, the client-side code is operable to retrieve the needed information from the client-side cache; wherein, if the client-side cache does not comprise the needed information, the client-side code is operable to register a request entry with the scheduler, wherein the request entry represents a request to retrieve the needed information from the server computer (see col.11, lines 42-55 and col.16, lines 21-32).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Braddy within the system of Phaal and Emens by implementing a cache in the client side within a client-server offline system because this allows for needed information to be accessed more quickly, rather than always having to connect to the server via a network. Caches are well known in the art for increasing operation speed.

As per *claims 11 and 41*, Phaal further teaches wherein the client-side code is operable to register a synchronization entry with the scheduler; wherein the synchronization entry represents a synchronization request to be sent by the client-side code to the server computer; wherein the scheduler stores the synchronization entry until a network connection is available for use by the client-side code; wherein the scheduler notifies the client-side code that the network connection is available; and wherein the client-side code is operable to send the synchronization request to the server computer and use the synchronization request results returned by the server

computer to synchronize at least a portion of the client-side cache with the server-side

database (see col.9, lines 51-59 and col.12, lines 27-34).

Phaal and Emens do not teach wherein the server-side code maintains a

server-side database. Braddy teaches wherein the server-side code maintains a

server-side database (see Fig.1 #16, #18, #20, & #22; and col.1, lines 57-60).

It would have been obvious to a person of ordinary skill in the art at the time the

invention was made to employ the teachings of Braddy within the system of Phaal and

Emens by implementing a database on the server-side within a client-server offline

system because most servers perform some function or task such as retaining,

updating, or executing of data which require large amounts of repository, therefore it is

well known in the art that databases are often and frequently used by servers to

maintain executable programs or data file frequently accessed remotely.

As per *claims 13 and 43*, Phaal further teaches wherein said client-side code

registering a synchronization entry with the scheduler comprises the client-side code

informing the scheduler of a synchronization callback routine to associate with the

synchronization entry; wherein the scheduler stores information identifying the

synchronization callback routine; wherein the scheduler invokes the synchronization

callback routine when a network connection is available for use by the synchronization

callback routine (see claim 2 and 32 rejection above).

As per *claims 14 and 44*, Phaal further teaches wherein said client-side code

registering a synchronization entry with the scheduler further comprises the client-side

code informing the scheduler of a time to invoke the synchronization callback routine

(see col.2, lines 62-64); wherein the scheduler stores information identifying the time to invoke the synchronization callback routine (see col.4, lines 48-53); wherein the scheduler invokes the synchronization callback routine when the specified invocation time occurs and a network connection is available for use by the synchronization callback routine (see col.4, lines 59-61).

As per *claims 15, 16, 45, and 46*, Phaal further teaches wherein, upon completing the synchronization of the at least a portion of the client-side cache with the server-side database, the synchronization callback routine informs the scheduler of a time to re-invoke the synchronization callback routine; wherein the scheduler stores information identifying the time to re-invoke the synchronization callback routine; wherein the scheduler re-invokes the synchronization callback routine when the specified re-invocation time occurs and a network connection is available for use by the synchronization callback routine; wherein the re-invocation time specified by the synchronization callback routine is calculated based on the amount of synchronization that the synchronization callback routine performed (see col.6, lines 25-30; col.9, lines 51-65; and col.12, lines 27-30).

As per *claims 17 and 47*, Phaal further teaches wherein a request entry and a synchronization entry are simultaneously registered with the scheduler (see col.12, lines 32-33); and wherein the scheduler prioritizes the request entry over the synchronization entry with regard to notifying the client-side code that a network connection is available (see col.4, lines 59-61).

As per *claims 19 and 49*, Phaal and Emens teaches all the limitations except wherein the server side of the client/server application comprises a first server computer; wherein requests sent by the client-side code are received by the first server computer and passed to a second server computer; wherein the second server computer carries out the request. Braddy teaches wherein the server side of the client/server application comprises a first server computer; wherein requests sent by the client-side code are received by the first server computer and passed to a second server computer; wherein the second server computer carries out the request (see Fig.4).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made, to employ the teachings of Braddy within the system of Phaal and Emens by implementing a second server to carry out requests within a client-server offline system because most providers of information consists of many servers for many different reasons. One such example, which is known and used in the art, is a gateway, router, or load server (first server) which routes to other plurality of servers based on various methods such as load balancing or request-categorization.

As per *claims 21 and 51*, Phaal and Emens teaches all the limitations except wherein the scheduler component is a component constructed according to a software component specification from the group consisting of COM, CORBA, and JavaBeans. Braddy teaches of CORBA, and JavaBeans (see col.4, lines 12-17 and col.9, lines 2-7).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Braddy within the system of Phaal and

Emens by implementing a scheduler component constructed according to a software

component specification from the group consisting of CORBA, and JavaBeans within a

client-server offline system because these programs or applications are industry

standards and widely used, which would allow for the scheduler to be easily integrated

and implemented in many different current systems.

9.   *Claims 26-27 and 56-57* are rejected under 35 U.S.C. 103(a) as being

unpatentable over Phaal (US 6,006,269 A) and Emens et al. (US 6,934,735 B1), and

further in view of Swanson et al. (US 6,112,183 A).

Although Phaal teaches an application enabling requests while the client

computer is disconnected from the Internet, Phaal and Emens do not explicitly teach

wherein the Internet application is a healthcare application; and wherein the application

is a healthcare application enabling healthcare workers to file health insurance claims.

Swanson teaches wherein the Internet application is a healthcare application

(see col.3, lines 6-10); and wherein the application is a healthcare application enabling

healthcare workers to file health insurance claims (see col.4 lines 42-56).

It would have been obvious to a person of ordinary skill in the art at the time the

invention was made to employ the teachings of Swanson within the system of Phaal

and Emens, by using the scheduler within the client-server offline system in the

healthcare industry because healthcare facilities deal with many distributed healthcare

workers filing claims, thus to ensure proper filing on-line using the Internet to a central

repository, Phaal's system can reduce cost and increase productivity by reducing the

need for extra equipment (extra servers to handle the load) and by allowing the claims

to be filed while disconnected and processed when connection resumes.

Allowable Subject Matter

10.   Claims 12 and 42 are objected to as being dependent upon a rejected base

claim, but would be allowable if rewritten in independent form including all of the

limitations of the base claim and any intervening claims.

The following is a statement of reasons for the indication of allowable subject

matter: Phaal (US 6,006,269 A), Emens et al. (US 6,934,735 B1), and Braddy (US

6,141,759 A) does not explicitly disclose, teach, or suggest wherein, in response to

receiving the synchronization request, the server computer is operable to return a list of

new items to add to the client-side cache, a list of items to delete from the client-side

cache, and a list of items to update in the client-side cache as recited in claims 12 and

42.

Conclusion

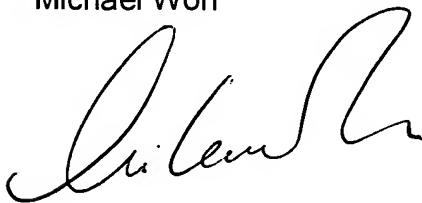11.   Claims 1-11, 13-41, and 43-60 have been rejected and remain pending.

12.   Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Y. Won whose telephone number is 571-272-3993. The examiner can normally be reached on M-Th: 7AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on 571-272-4006. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Won

March 6, 2006

SALEH NAJJAR
SUPERVISORY PATENT EXAMINER